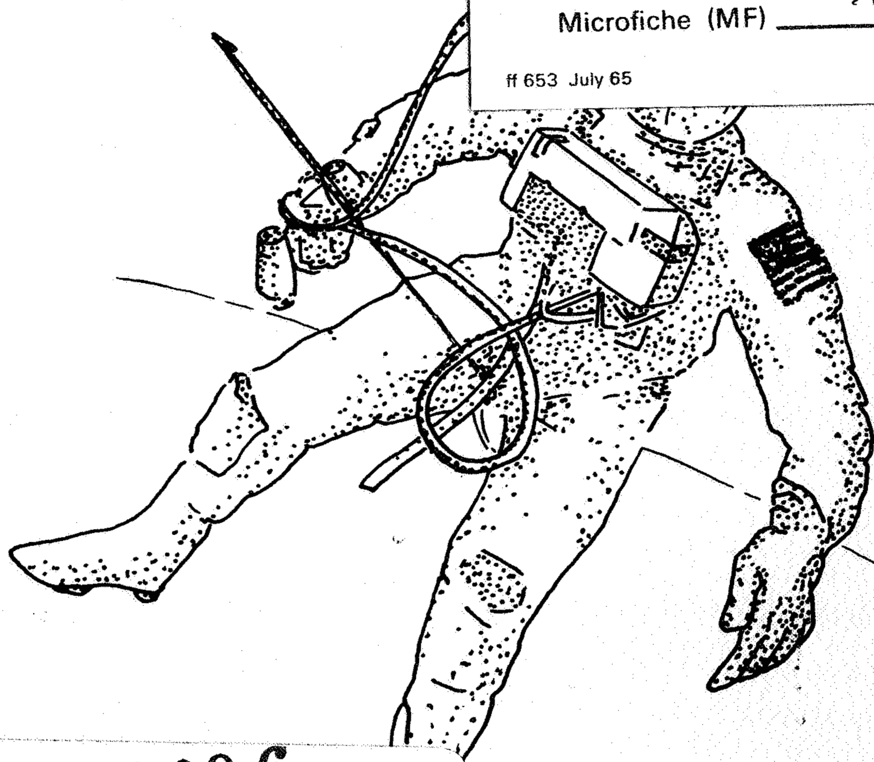
NAS 9-996 REAL TIME COMPUTER COMPLEX

Project Gemini Final Report

SUMMARY



IBM FEDERAL SYSTEMS DIVISION
HOUSTON, TEXAS

NASA CR 92180

Project Gemini Final Report

IBM

NAS 9-996 Real Time Computer Complex   Federal Systems Division   Houston, Texas

- - - -

PROJECT GEMINI FINAL REPORT

SUMMARY

- - - -

Submitted to

National Aeronautics and Space Administration
Manned Spacecraft Center
Houston, Texas 77058

Contract No. NAS 9-996

Federal Systems Division
International Business Machines Corporation
1322 Space Park Drive
Houston, Texas 77058

FOREWORD

This report is submitted to the National Aeronautics and Space Administration as a supplement to the IBM RTCC Gemini Final Report. It is intended to serve as a summary of that report and is designed for the reader who desires a general idea of the scope and nature of IBM's contribution to the Gemini Project. For detailed study of the technology and development of Gemini RTCC systems, the reader is referred to the above report and the documentation mentioned therein.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

INTRODUCTION

On October 15, 1962, IBM was informed that it had been selected as contractor for the Real Time Computer Complex (RTCC) for the Gemini and Apollo space projects. The RTCC would be part of the control center at the Manned Spacecraft Center being constructed near Houston by the National Aeronautics and Space Administration. As in Project Mercury, IBM was pleased to call on its resources and experience to work with NASA and accept the challenge of developing the new methods and technology necessary to complete the project and advance the leadership of the United States in the exploration of outer space.

For Project Mercury, IBM had provided equipment, program development, and operations support for real time control of the first manned space flight missions. This experience was of great value in preparing the initial organization and objectives of the RTCC.

Project Gemini was planned for the development and operation of spacecraft capable of extended orbital space flights and orbital rendezvous. These missions involved liftoff and insertion using a Titan II booster; low, near-circular orbit; return using aerodynamic lifting reentry techniques; and abort capability at any point during the mission. Certain missions were planned to include rendezvous and docking with an Agena target vehicle launched in advance using an Atlas booster. Rendezvous plans were later modified to include a rendezvous between two Gemini craft during the historic Gemini 7/6 mission.

Support for the Gemini project was provided by the Ground Operation Support System (GOSS) as it had been for Mercury. IBM's contract included two elements of the GOSS, the RTCC and the Launch Trajectory Data System.

The RTCC is the ground-based data processing system for the manned space flight program. Its function is to execute real time computations required for successful accomplishment of NASA program objectives including monitoring and support of missions, simulations and training for the Gemini flights.

IBM was to design, develop, implement, and operate the RTCC. This responsibility included furnishing computer equipment, associated peripheral equipment, and programming systems. Also included were systems studies, mission and mathematical analysis, special equipment engineering, equipment and program testing, maintenance, operation, and documentation for the RTCC.

The Launch Trajectory Data System was that portion of the Launch Data System that controlled and transmitted trajectory and guidance data between the launch monitor facilities and the Mission Control Center during the period from pre-launch through orbital insertion of the spacecraft or the target vehicle. During simulation, it controlled and transmitted trajectory and guidance data between Houston and the Gemini Mission Simulator at Cape Kennedy. The work on the Launch Trajectory Data System was a continuation of a previous contract with NASA.

The original RTCC contract, signed at the completion of negotiations on July 12, 1963, called for expenditures of $36 million over a period extending to August 31, 1965. The responsibilities of NASA and IBM under the contract were described in a document known as the RTCC Statement of Work. Throughout the course of the project, the contract and statement of work were periodically modified to reflect changes in scope of the work to be performed and additional funding. At the completion of the project in December 1966, Gemini expenses for the RTCC totaled an estimated $43 million with approximately 1.5 million man-hours of effort. The size of the organization grew to 544 at the time of the Gemini 7/6 rendezvous. Remaining Gemini missions plus initial Project Apollo development and support caused an increase to 608 by the end of Gemini 12. Slightly less than two-thirds of the organization was involved in program development. One fourth of the personnel were engaged in maintenance, operation, and engineering of equipment. The remaining personnel supplied various support services. Additional technical and administrative staffs were applied in Houston and at other locations in support of the project throughout its life. In this manner all the facilities of the IBM Corporation were made available, both generally and specifically, to the RTCC project.

IBM began operations with a small staff in temporary locations in Houston in October 1962. Planning of program and equipment development was done in close conjunction with NASA personnel. Programming began at Houston while special equipment design continued in Kingston, New York. Since progress depended upon agreement as to requirements, and requirements were subject to change as technology and design progressed, many difficulties were encountered in the early stages of development. It is due to the excellent cooperative and flexible attitudes on the part of NASA and its contractors that these difficulties were overcome.

Early in 1963, an interim facility was established in IBM's temporary headquarters in Houston. Three IBM 7094 and two IBM 1401 systems with an interim Communications Processor and display console facilities were installed. This proved to be of great assistance in evaluating early programs and interfaces and aided in making design decisions.

By November 1964, IBM personnel were moved to facilities at Clear Lake City, adjacent to the Manned Spacecraft Center, and two months later the RTCC was monitoring the unmanned Gemini-Titan 2 flight. A final test of the RTCC was made on March 23, 1965, with full-scale monitoring of Gemini 3, the first manned mission of a Gemini spacecraft.

The NASA decision to shift permanent control of manned missions from Cape Kennedy to Houston came in April 1965. On June 3, RTCC provided primary mission control for Gemini 4 and presented data for NASA flight controllers to make real-time decisions during the entire four-day flight. The original plan had been to begin full mission support with Gemini 6, which actually took place in December.

Two record-setting long duration missions followed, with RTCC providing critical data in real time to controllers, first for eight days, then for fourteen, in support of Gemini 5 and 7.

During the Gemini 7 mission, RTCC computers processed information over a two-week period, including three days in support of the Gemini 6 rendezvous with Gemini 7. This required the IBM 7094 computers to process data from two spacecraft simultaneously.

Throughout 1966 the RTCC supported the last five Gemini missions, gaining valuable experience in development of rendezvous systems while planning and supporting early Apollo mission systems.

THE RTCC SYSTEM

The RTCC was in reality a group of systems developed in fulfillment of the RTCC contract. Operationally there were two major systems, the Mission Operational System and the Ground Support Simulation Computer system, plus several smaller ones.

The Gemini Mission Operational System provided data processing before and during a Gemini mission. Elements of this system were computers and special equipment plus control and operational programs.

The Ground Support Simulation Computer system was the central part of the Simulation, Checkout and Training System (SCATS). SCATS was used to carry out the pre-mission simulation and training exercises for Mission Operational Control Room teams, remote site teams, and astronauts. The GSSC system provided several varied functions within SCATS.

These systems plus others, such as the Operational Readiness and Confidence Testing system, the Dynamic Network Data Generator system, and the Compiler Operating System will be described in more detail in later sections. In most cases the systems described are those used for support of the Gemini 12 mission and were the most complicated systems developed.

The major RTCC systems were "real time" systems. A real time system is a data processing system that accepts and processes data as it occurs (or within very short time limits). Obviously this capability is necessary during a manned space mission to continuously provide current status information and react instantly to the decisions of flight controllers and astronauts.

A primary objective of the Gemini program was the safe return and recovery of the flight crew. IBM sought to accomplish this objective through the use of appropriate planning of programs and operational procedures and thorough detailed testing procedures. No mission was run without complete testing of all elements of the programming system supporting it. In addition, program system design incorporated redundant functions to enable flight controllers to always have an alternate method available to accomplish any activity should a malfunction occur. Certain phases of missions were identified as critical phases, such as an abort during a launch. Exhaustive efforts were expended to assure proper system performance during these phases. In addition to the IBM standard internal checking built into the hardware, an extra computer system duplicated computations during missions to provide a backup system in case of failure of the primary system.

In terms of time, many programming systems were developed in the RTCC. Since each Gemini mission was different (new flight plan, new maneuvers, equipment changes, different duration), mission control requirements were significantly modified for each mission. These requirements were provided to IBM for future missions, which meant that a number of mission programs were under development simultaneously. Thus, for each mission, IBM was responsible for the delivery of a mission program system, a simulation program system, and certain support programs. While these were usually modifications to systems previously developed, nevertheless, each represented considerable planning, coordination, and programming effort.

MISSION OPERATIONAL PROGRAM SYSTEM

This section describes the Mission Operational Program System used during the Gemini 12 mission. It presents the program as an integrated whole functioning in real time during various phases of the mission. Also included is a discussion of the control programs which were used to achieve the integration of the subsystems.

The Mission Operational Program System was a one-half-million-word real time program that monitored telemetry and radar inputs from around the world and responded to the manual requests of flight controllers located in the Mission Control Center in Houston, Texas. The program presented data to the flight controllers via panel indicators, X-Y paper and rear projection plotboards, and a television system capable of presenting any 28 of 300 possible displays at any given time. It was designed to aid the flight controller in understanding the current and future vehicle environment and to aid his planning of upcoming mission activities by rapidly preparing and presenting displays in an easily readable form. Major functions of the program included:

a. Recommending a spacecraft liftoff time based on the Agena orbit.

b. Processing high-speed radar data during a launch and preparing displays to describe the launch trajectory characteristics and the launch to orbit insertion conditions.

c. Processing low-speed radar data to determine the present vehicle position and to predict vehicle trajectory.

d. Aiding in the planning of midcourse, rendezvous, special purpose and retrofire maneuvers based on predicted trajectory.

e. Predicting the spacecraft's impact point during launch, abort, and reentry based on flight controller input or computer-determined retrofire conditions.

f. Preparing commands for the Gemini and Agena onboard computers to control maneuvers and system functions.

g. Processing input telemetry data for the Titan, Gemini, and Agena and displaying it to show trends, summaries, and histories of vehicle parameters.

h. Preparing data to aid on-site and flight controller personnel in the planning of tracking and recovery operations.

i. Preparing displays to aid in planning onboard experiments.

For development and testing purposes the Mission System was divided into eight subsystems:

Agena Launch

Gemini Launch

Orbit

Trajectory Determination

Mission Planning

Telemetry

Digital Command System

Reentry

These subsystems were executed as required by events during mission phases. Each subsystem was composed of several programs working together to perform subsystem processing. The following sections describe these subsystems.

MISSION SYSTEM ELEMENTS

Agena Launch Subsystem

The Agena Launch Subsystem was used to monitor the trajectory of the Agena Target Vehicle from its liftoff until orbit was attained. Tracking data was transmitted from the Impact Predictor Computer at Cape Kennedy and from the Bermuda radar site. The subsystem validated this data, then calculated and smoothed the trajectory path. Position, altitude, and velocity vectors and time were displayed to flight controllers during powered and free flight. The actual trajectory was plotted against the nominal (predicted) trajectory, and a Go/No-Go recommendation was computed and displayed in the Mission Operational Control Room.

Gemini Launch Subsystem

The Gemini Launch Subsystem was used to monitor that period in the mission from Gemini liftoff until either the Abort or Orbit phase was entered. The purpose of the subsystem was twofold:

1. To monitor the booster and spacecraft radar track for a possible non-nominal condition in the trajectory. Such a condition could lead to a flight controller decision to initiate corrective action or to abort the mission.
2. To calculate the best available spacecraft position and velocity vectors for use in orbit determination or for impact prediction in case of an abort.

High-speed radar tracking data was received for two vehicles, the booster and the spacecraft. These data sources, as in the Agena Launch Subsystem, were validated and edited by the subsystem.

During powered flight, trajectory parameters and impact predictions were calculated every half-second. These were available for display and trajectory data for plotting against the nominal flight path. During free flight (after sustainer engine cutoff), similar processing continued, and in addition, Go/No-Go and time-to-fire parameters were calculated and displayed. The time-to-fire calculations determined a time to initiate retrofire taking into account possible primary and contingent recovery areas.

Orbit Subsystem

The Orbit Subsystem computed predictions of orbits to be traversed by the space vehicles. These predictions had to be recomputed whenever events occurred that altered or improved the previous prediction. Ephemeris tables were maintained reflecting executed and planned maneuvers to enable flight controllers to view results of computations for the current revolution and up to 22 revolutions in the future. (See Figure 1.)

The subsystem provided all computations necessary for flight control monitoring and decision-making relating to both the orbiting Agena and Gemini spacecraft in addition to processing for a continuing evaluation of planned or emergency reentry.

Using the latest ephemeris prediction as the foundation for calculations, the subsystem displayed information on request which defined orbital characteristics for present and future revolutions. Additional displays provided vital data pertaining to radar tracking coverage, reentry planning, celestial and ground points, and periods of daylight and darkness. The primary device for displaying data was the controller console digital-television monitor. Other display devices driven by the Orbit Subsystem were the X-Y and projection plotboards. These provided large

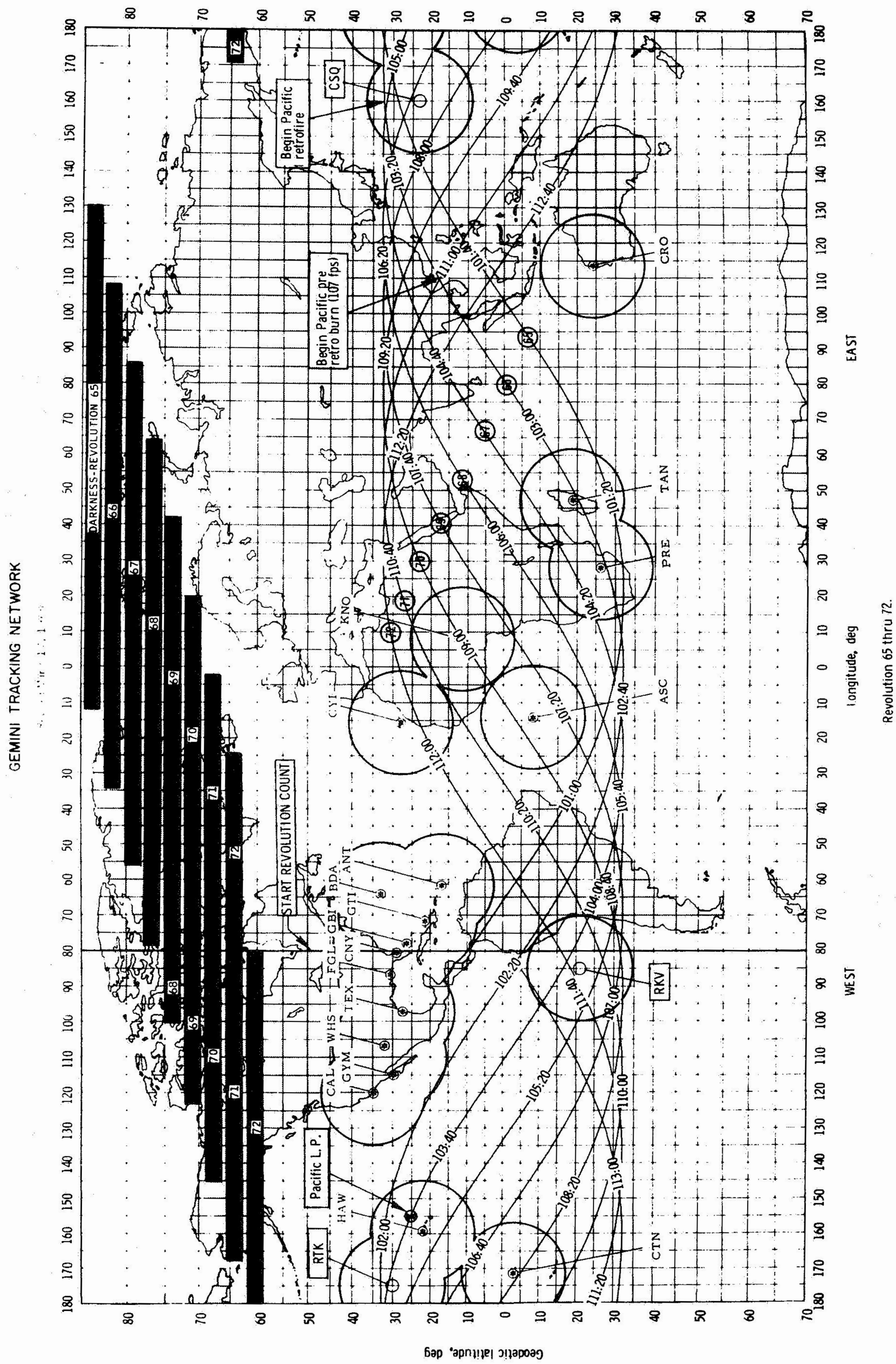


Figure 1. Typical Groundtracks of Orbits

analog plots for group viewing and indicated present position, velocity and height, as well as the relative orbital characteristics for the Agena and Gemini vehicles.

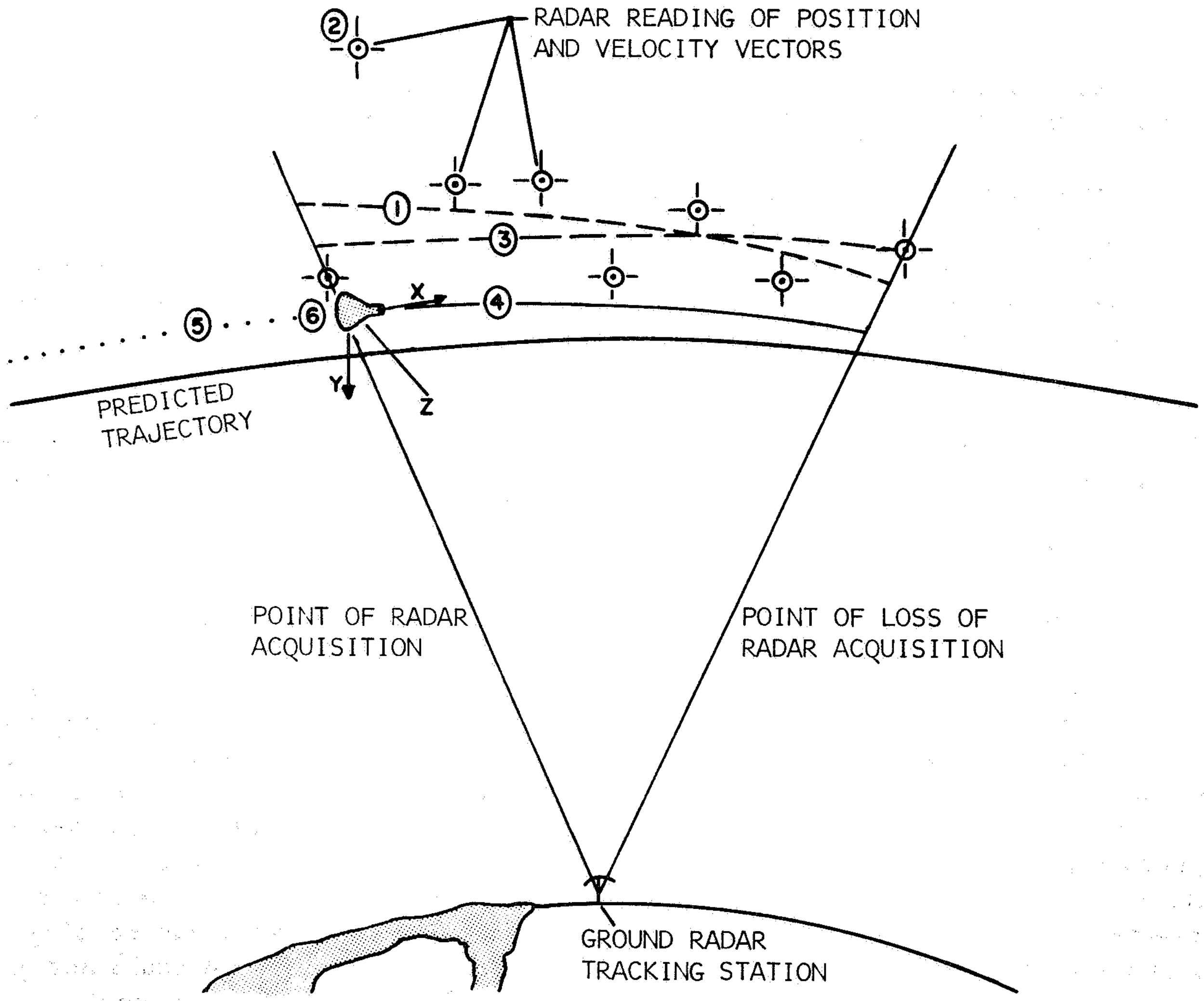
A further purpose of the Orbit Subsystem processing was to compute, format, and provide for teletype transmission of the acquisition data to inform remote radar stations when and where orbiting vehicles would be in range of their particular radar. Department of Defense stations also received inertial position and velocity data formatted for direct processing in on-site computers.

The ephemeris tables calculated by the Orbit Subsystem were valid as long as no changes in mission plans took place and as long as the spacecraft remained on its planned course. Frequent orbit computations were necessary, however, since changes to the mission plan were common, and trajectory calculations frequently determined that the spacecraft position deviated from plans.

Trajectory Determination Subsystem

The Trajectory Determination Subsystem determined the position of the spacecraft or Agena target vehicle at a given point in time and evaluated and compared data from the remote tracking stations (see Figure 2). The subsystem received low-speed radar readings of position and velocity vectors taken every six seconds by a remote tracking site and performed calculations to determine the probable trajectory path across that tracking site plus corrected velocity and position vectors at a given point in time. Visual displays enabled flight controllers to compare the path to a precalculated trajectory path based on previous readings and calculations. Several options were open to the controller. He could accept or reject any or all of the radar readings from the site. He could request a backward projection of the trajectory from the latest reading to compare the results to those of previous radar sites. The controller could alter the calculations based on observations of the characteristics of a tracking station. If that station, for instance, had a history of increasing the altitude slightly for each reading, the controller could request the subsystem to "degrade" altitude in its calculations. The trajectory determinations were also useful to controllers in determining if a recalculation of the orbit ephemeris tables was needed.

The Trajectory Determination Subsystem employed an iterative differential correction process using the Runge Kutta integrator and a covariance matrix based on past observations to determine the position and velocity vectors of the space vehicle. A Cowell integration routine was used to project trajectory paths over more than one tracking station. This statistical method incorporated vehicle maneuvers, trajectory parameter (vector) uncertainties, and the trajectory prediction errors manifest over extended time periods.



1. Probable trajectory across a tracking station is calculated based on radar readings of positions.
2. Any apparent unreliable readings may be rejected at the discretion of the flight controller.
3. Adjusted probable path.
4. Further adjustment due to station characteristics (e.g., altitude degraded).
5. Backward integration based on adjusted probable path permits comparision to previous stations' probable paths.
6. Corrected trajectory showing position and velocity vectors at the point of radar acquisition for the site.

Figure 2. Trajectory Determination

The trajectory estimation process could incorporate the information of a new batch of radar observations in about one-seventh the time required by classical differential techniques used in Project Mercury. The method was consistently accurate and successfully supported all missions from Gemini 4 through 12, and represents a significant contribution to space mission technology.

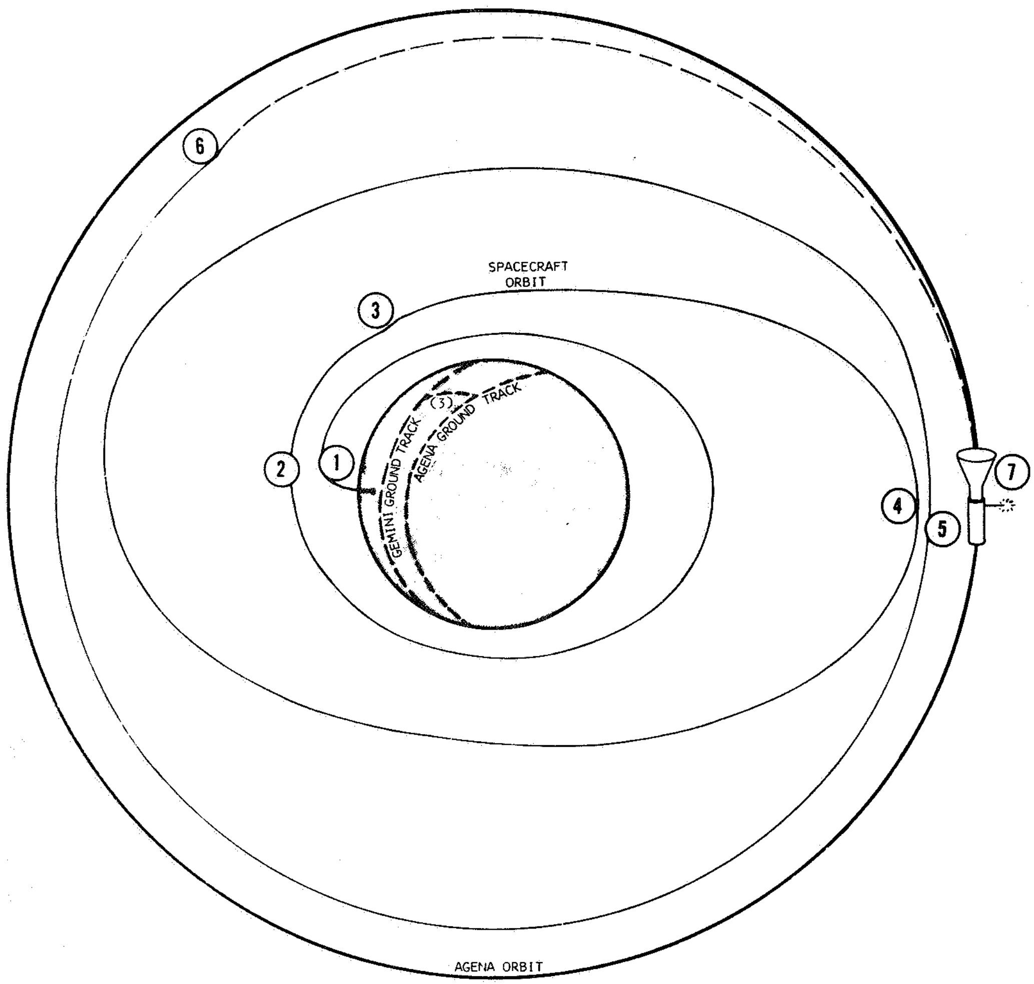
Mission Planning Subsystem

The Mission Planning Subsystem assisted in planning, performing, and recording the effects of spacecraft and Agena thrusting maneuvers during orbit. The subsystem planned and controlled all maneuver computations for Gemini rendezvous missions. It also provided computations to target the spacecraft to the Agena orbit, thus giving NASA flight controllers a real time launch planning capability (see Figure 3).

The programs were designed to give the flight controller maximum flexibility and control of all maneuver processing as well as to provide a maximum of automatic maneuver update logic. "Maneuver control" was achieved by a combination of manual and automatic control methods. Manual control of a mission plan was provided by inserting data through a manual input device. This input data resulted in the generation of a single maneuver or group of maneuvers. Existing mission plan maneuvers were replaced, deleted or altered at the request of the flight controllers. Executed maneuvers could be changed (confirmed) to agree with the actual manner in which the maneuver was performed. Vehicle characteristics, such as weight and fuel remaining, were updated by manual input. Automatic control was provided in the computer programs by checking mission plan ground rules, controlling mission plan updates, detecting abnormal conditions and requesting the computation of an alternate plan when necessary. Manual control inputs could be overridden or rejected when subjected to automatic control testing.

The Mission Planning Subsystem consisted of five program units: Launch Planning, General Purpose Maneuver Computations, Docking Initiate Computation (Rendezvous Maneuver Planning), Two-Impulse Computation, and Maneuver Control Processing.

The Launch Planning Program computed Titan guidance parameters necessary for the spacecraft to be inserted into an orbit whose orientation and energy relationships with the Agena target orbit were such that rendezvous could be achieved. The program was executed prior to launch of the Gemini vehicle.



RTCC calculates Gemini launch time and azimuth and maneuvers necessary to rendezvous with Agena at point 7. Point 7 is predicted Agena position, velocity and time based on the Agena orbit determination. The scale of the orbits is greatly exaggerated for clarity.

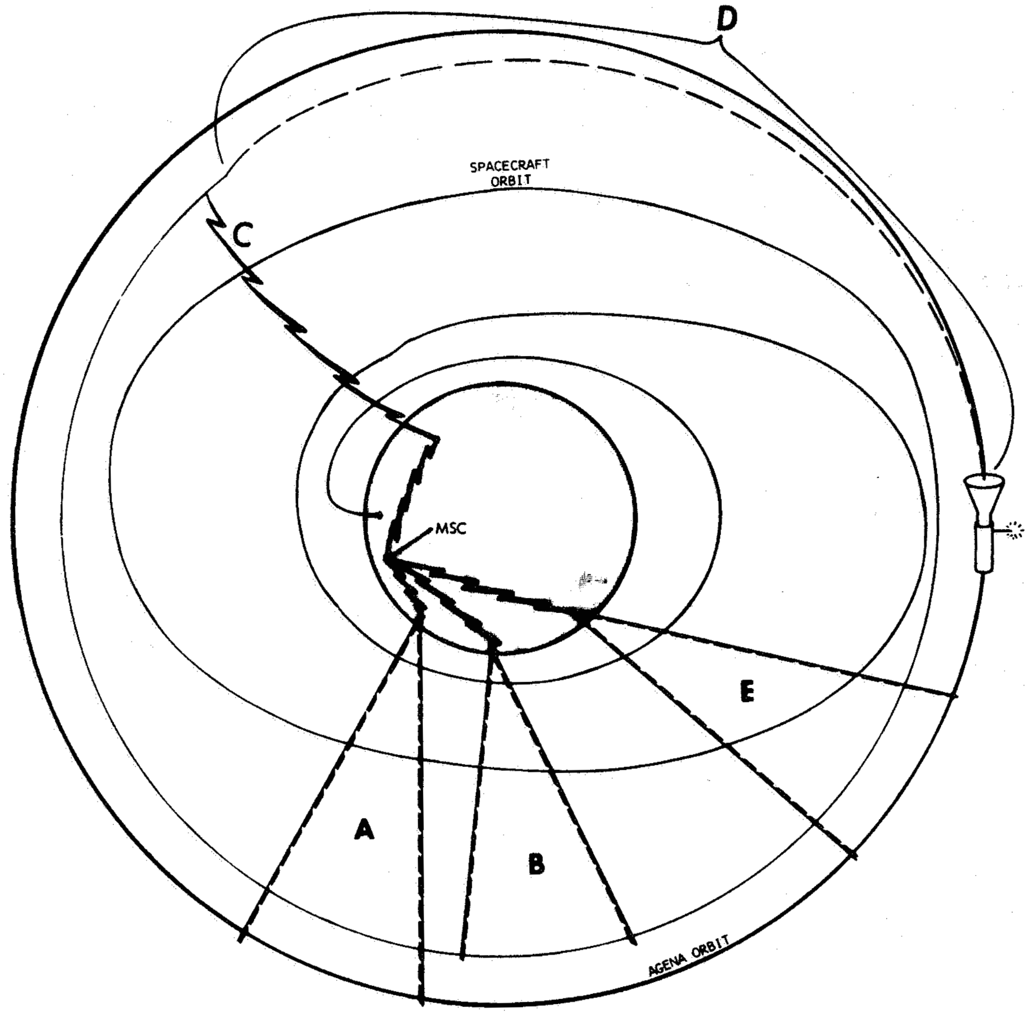
1. Gemini orbit insertion.
2. A height adjust maneuver is planned to raised orbit apogee.
3. A plane change maneuver will put the spacecraft into the same orbit plane as the Agena, creating coincident ground tracks.
4. A phase adjust maneuver at second apogee will effect the required spacecraft catch-up rate.
5. A maneuver is made to attain orbit that is coelliptic with Agena.
6. The Terminal Phase Initiate maneuver transfers the spacecraft to the planned point of rendezvous.
7. The Terminal Phase Break maneuver puts the spacecraft into the Agena orbit at the rendezvous point.

Figure 3. Mission Planning

The General Purpose Maneuver aided the Gemini flight controllers in determining what effect certain changes in velocity, pitch and yaw would have on either the Gemini or Agena orbit, or what magnitude of maneuver was needed to obtain a specified orbit. The two types of general purpose maneuvers were computed for either vehicle through manual entry requests. In the first type, the changes in velocity, pitch, yaw and a place in the orbit for the maneuver to be performed were input by the flight controller, and the program computed the resulting orbit. The other type of maneuver required as input a desired change in some orbital parameter, such as apogee or perigee, and the program computed the maneuver necessary to obtain the desired orbit. The output of the program was a display of the vehicle name, "vector identification", time, change in velocity, pitch and yaw of the maneuver, the resulting orbital parameters, height at apogee, height at perigee, eccentricity, and inclination as well as a set of orbital elements before and after the maneuver which could be used in trajectory determination. The "vector identification" referred to was an identifying symbol for the tracking station and orbit which supplied the velocity and position vector upon which the maneuver computations were based.

The Docking Initiate and Two-Impulse Computation programs mainly were used to assist in maneuvering the spacecraft to a position relative to the Agena so that terminal phase maneuvers could take place under control of the onboard computer and guidance system. The Docking Initiate program computed a series of maneuvers that would produce the required coelliptic trajectories to achieve terminal phase maneuvers for rendezvous at a given point.

The Two-Impulse Computation Program gave the Gemini flight controllers a means for computing a transfer trajectory necessary to bring about a desired orbit change. The program computations were initiated by a manual entry in terms of chaser and target vehicles. The Gemini spacecraft could be defined as the chaser vehicle and the Agena as the target vehicle, or vice-versa. The program computed a chaser transfer initiation maneuver needed to intercept a point in space that was defined in terms of the target vehicle. The results of the computations were available for display to the flight controllers in one of two formats. The first, the "impulsive" format, considered the maneuver to have taken place in two specific instantaneous actions. "Impulse" times, velocity increments, pitch and yaw angles were displayed. The "finite-burn" format considered the firing of thrusters in terms of the time they were actually firing. Finite-burn data displayed were begin-burn times, burn durations, velocity increments, and pitch and yaw angles. In addition, daylight-darkness information associated with both maneuvers were displayed. Range, range-rate, and azimuth approach data were displayed at transfer initiation, begin-burn, and at three other prior times.



A. Ground tracking station tracks Agena, receives telemetry data and transmits radar and telemetry data to MSC.
B. Ground tracking station tracks spacecraft, transmits radar and telemetry data to MSC, and transmits RTCC output data to the spacecraft.
C. RTCC Digital Command Subsystem computes data for spacecraft onboard computer and begins transmission to the spacecraft.
D. Onboard computer controls spacecraft during final maneuvering while RTCC monitors maneuvers.
E. After rendezvous, RTCC updates orbit trajectory and ephemeris based on Agena and spacecraft position.

Figure 4. Telemetry and Digital Commands During Rendezvous Maneuvers

The Maneuver Control Processing program was responsible for processing various inputs related to maneuver control and/or trajectory updates. The program controlled communications between the Mission Planning Subsystem and the Trajectory and Orbit Determination subsystems to insert corrections in the trajectory and ephemeris tables. It monitored mission activity and flight controller inputs to enforce predetermined mission planning ground rules. For example, a request to initiate a maneuver prior to a maneuver already planned was illegal and was rejected by the program. The program reflected changes in conditions, such as changes in spacecraft weight or confirmation of maneuvers immediately after they took place, and refinements of maneuvers based on newly calculated trajectory vectors. The program also maintained a summary maneuver table, which contained data on all of the mission maneuvers.

Telemetry Subsystem

The basic function of the Telemetry Subsystem was to reduce the mass of electronic signals emanating from the Titan, Agena, and Gemini vehicles to a form readily interpretive and immediately accessible to the flight controllers. The subsystem acquired the data from the vehicle, tested it for validity, and converted it from its raw input form to engineering units. The subsystem then performed any needed calculations and displayed the results. Figure 4 illustrates telemetry processing resulting from the Mission Planning functions. Some other measurements processed by telemetry programming included fuel quantities, gas pressures, electrical system voltages, attitude control status, cabin pressure/temperature, temperature/pressure of astronaut suits, and medical indicators of the astronauts.

A variety of input formats and data rates was handled by the subsystem, and several types of display formats were produced. These displays included such things as history tables, trends displays, Titan plots, and Agena schematics. Telemetry data was also recorded on magnetic tape and listed periodically or as requested for in-flight analysis.

Digital Command System Subsystem

The Digital Command System (DCS) Subsystem was used to generate digital commands for transmission to the Gemini onboard computer and the Agena memory (see Figure 4). The commands sent to the onboard computer were used to generate an onboard ephemeris, control guidance during reentry, and control orbital maneuvers. The quantities transmitted were used as input to the onboard computer programs, then initiation of the onboard computations was determined by the astronaut.

Since the Agena memory had no processing capabilities as in the Gemini onboard computer, the commands had to be formatted in the RTCC in a form acceptable to the Agena memory scanner. The commands were loaded into the sixty-four-row memory, and each row was scanned once every second. When the time tag of any command matched the time in the Agena clock, that particular command was executed. The command loads were used for ground control to maneuver the Agena, control the telemetry, and turn the beacons on and off.

Reentry Computations Subsystem

Reentry processing consisted of a collection of programs which provided information to plan reentry and select recovery areas based on existing orbital conditions, and another collection of programs that provided nominal reentry information 25 minutes prior to planned retrofire with monitoring for actual reentries or aborts.

A time-to-fire supervisor predicted retrofire times based on the current trajectory and desired maneuvers to splash down in given target areas.

These predicted times were calculated for the upcoming primary and contingency recovery target area for each trajectory re-determination, and they could be calculated for a specified target area when requested by a flight controller. This manually-entered target area could be up to 22 revolutions from the current spacecraft position, and could be designated as either primary or contingent.

In some cases, a retrofire was not practical for a given target area during a particular revolution. The time-to-fire supervisor then calculated a retrofire time that would direct the spacecraft as close as possible to the desired area. It also calculated the distance in latitude and longitude to the actual impact point. Hence, the program gave a measure of the advisability as well as possibility of using a given target area for reentry and recovery.

PROCESSING FUNCTIONS DURING MISSIONS

Each Gemini mission was based on eleven phases of flight (see Figure 5). A phase defined which processing would be automatically executed, which displays would be automatically updated, which data sources would be permitted to enter the computer, and the frequency of the display updating. To do this, certain subsystems were activated during each phase of the mission. The phases were:

Prelaunch 1, Agena Prelaunch

Launch 1, Agena Launch

Prelaunch 2, Gemini Prelaunch

Launch 2, Gemini Launch

Abort

Midcourse

Terminal

Post-Docking

Pre-Retrofire

Reentry

Recovery, Agena Orbit

Prelaunch 1 Phase

The Prelaunch 1 phase began when the system was loaded into the computers. During Prelaunch 1, the system processed high-speed telemetry data from Cape Kennedy for the Titan booster and for the Agena and Gemini spacecrafts. High-speed radar was processed from the launch sites to maintain real time displays in the Mission Control Room.

Launch 1 (Agena Launch) Phase

Launch 1 began at Agena liftoff and continued until orbit entry. During this phase, telemetry processing for the three systems (Agena, Titan, Gemini) continued. In the Mission Control Room, altitude and velocity data were plotted and displayed against nominal values during powered flight, and an orbit Go/No-Go recommendation was displayed during free flight based on a comparison of the actual and the desired Agena cutoff conditions.

Prelaunch 2 Phase

Prelaunch 2 began at Agena orbit and terminated at Gemini liftoff. During this phase, Titan and Gemini high-speed telemetry data was processed, and displays were updated every second. Agena low-speed telemetry was processed and displayed as received. Agena low-speed radar was received from the worldwide tracking network and stored for later processing. Agena trajectory displays

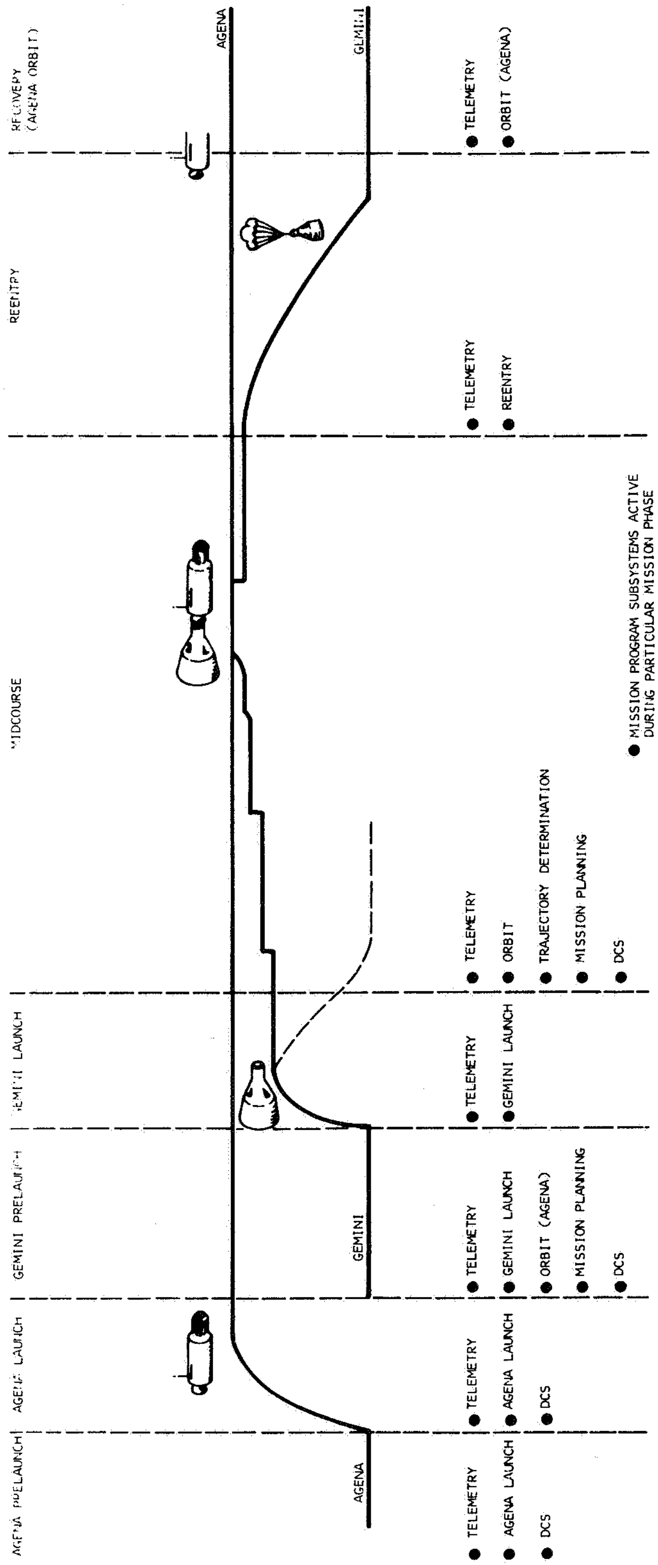


Figure 5. Mission System Phases

were updated on a 12-second cycle during this period. Agena acquisition data (data to direct radar pointing) was sent to each radar site approximately 25 minutes prior to acquisition. Planning of spacecraft liftoff time, desired rendezvous maneuvers, and maneuvers to shape the Agena orbit were accomplished by the Mission Planning Subsystem.

Launch 2 (Gemini Launch) Phase

Launch 2 began at Gemini liftoff and continued until the Flight Dynamics Officer set a switch to designate abort or orbit. During this phase, processing of high-speed telemetry data continued for the three systems. Titan high-speed telemetry data was processed until 8 minutes after liftoff.

During powered flight, the Trajectory Subsystem displayed the present position, altitude, and velocity and compared flight path and velocity relationships to the nominal values. These displays were updated every half-second. Impact points were predicted for use in the event of an abort during launch based on seat ejection, salvo retrofire, or controlled retrofire with minimum and maximum time delays.

During Gemini free-flight processing, the system displayed the vehicle's position, a Go/No-Go recommendation, the velocities to be added to achieve minimum and nominal orbital insertion, and the time-to-retrofire to land at an available predetermined recovery area. If the vehicle launch trajectory indicated an acceptable apogee height would be achieved but that reentry would occur in the first orbit, the Gemini Launch Subsystem recommended an "apogee kick" maneuver be performed in orbit to raise perigee and prevent reentry.

Abort Phase

Abort phase, if it had been used, would have begun when the Abort switch setting was recognized by the Gemini launch program. For seat eject (Mode I) and salvo retrofire (Mode II) aborts, high-speed radar was to be processed to determine the impact point. For controlled retrofire aborts (Mode III), the impact point was to be computed using the retrofire time and the vector obtained from the launch time-to-fire computations.

During Abort phase, displays indicating the impact point, landing time and other critical event times, and backup reentry guidance methods would have been updated on a six-second output cycle. Telemetry processing would have continued at a one-second output cycle while high-speed telemetry was being received.

Low-speed radar from Agena would have been collected after entry into Abort phase but not processed until directions to do so were given. The Abort phase terminated when an input was received directing the program to enter Recovery phase.

Midcourse Phase

Midcourse phase began when the Gemini vehicle achieved orbit and continued until a manual input of the time of retrofire was set by the Flight Dynamics Processing Controller.

During the Midcourse phase, the Orbit and Trajectory Determination subsystems produced position and velocity vectors which were used to maintain ephemeris tables of predicted orbits. Ephemeris tables were updated whenever events occurred that could alter the previous prediction. Telemetry and trajectory displays continued to be updated.

The Orbit Subsystem sent acquisition data to the tracking network sites approximately 25 minutes prior to the time the vehicles would be in view of a site. A test to determine the need to send data was made each time a vehicle passed through 20 degrees of longitude (approximately every 5 minutes).

A plan of the midcourse maneuvers required to achieve a desired phase and height relationship at a desired orbital position was developed by the Mission Planning Subsystem and was used to update predicted Agena and Gemini vehicle ephemerides. When a midcourse maneuver plan was reflected in the ephemeris, the program performed a simulation of the onboard computer processing and determined the Terminal phase maneuvers. The Terminal phase maneuvers were then also reflected in the predicted ephemeris. Detailed descriptions of the midcourse and terminal maneuvers were available to controllers on manual request. Planning orbit shaping maneuvers for either vehicle was accomplished and reflected in the predicted vehicle ephemeris.

Throughout this phase, the computer updated the time-to-fire to reenter and splashdown at the upcoming contingency recovery areas. The time-to-fire to reenter in any recovery area available within the next 22 revolutions could be obtained by manual request. Displays relating the vehicle groundtrack to earth-fixed points and the vehicle ephemeris to the ephemeris of other vehicles or celestial bodies were also available. Digital Command System data for loading the Agena and Gemini onboard computers was computed and transmitted to both vehicles prior to the Terminal phase.

Terminal Phase

Terminal phase began when the midcourse maneuvers had been completed and when the rendezvous maneuvers were being executed under onboard control. Processing during Terminal phase was the same as processing during Midcourse phase.

Post-Docking Phase

Post-Docking phase began upon receipt of an indication that the Gemini had docked with the Agena. Upon entry into Post-Docking phase, a current Agena vector was obtained from the Agena trajectory table and was used to update the Gemini predicted ephemeris. This logic assumed, because the Gemini maneuvers were being performed under onboard control, that the RTCC-computed ephemeris might vary slightly from that computed on board. Since the Agena trajectory should have been well established, and since the Gemini was known to be docked with the Agena, it was assumed that the Agena vector was the best estimate of the Gemini position and velocity. Processing during the Post-Docking phase was the same as during the Midcourse phase.

Pre-Retrofire Phase

Pre-Retrofire phase began when the Gemini vehicle had separated from the Agena vehicle. During this phase, maneuvers to shape the Gemini orbit to an optimum retrofire environment were planned.

Reentry Phase

Reentry phase began when the planned or actual retrofire conditions were manually input to the computer by the Flight Dynamics Processing Controller and continued until a manual entry was made to recycle to the Midcourse phase or to enter the Recovery phase.

As radar stations completed transmission, all observations for each radar station that were post-retrofire were made available to the reentry differential correction programs in the Trajectory Determination Subsystem. They updated the predicted reentry trajectory based on the observed data using the current input lift profile. Each time the predicted reentry trajectory was updated, acquisition data was sent to the radar sites that were to observe reentry.

Recovery Phase

The Recovery phase began when the Flight Dynamics Processing Controller entered a request to change to Recovery phase. Generally, the capsule had been sighted, and further Gemini processing was not desired. This phase was often refered to as the Agena Orbit phase because emphasis was placed on monitoring the Agena systems and planning Agena maneuvers to evaluate the maneuver system and to shape the Agena orbit

CONTROL PROGRAMS

Control programs provided the logic to permit initialization of programs for Gemini mission applications. They bridged the logic gaps between mission phases and mission subsystems; they aided in the control and output of data to the control room display systems; and they provided for program control via manual entry devices. The five control programs used during the Gemini missions are discussed below.

Mission Supervisor

The overall mission logic flow was controlled by the Mission Supervisor which monitored the input data and directed routing of it to the appropriate parts of the system. Major functions performed by the Mission Supervisor include:

1. Recognizing mission phase changes and notifying the appropriate phase supervisors of those changes.
2. Activating and deactivating routing of radar data according to mission phase and events.
3. Interrogating and routing Manual Entry Device inputs from mission controllers according to the message codes.
4. Interpreting and maintaining the status of the controller switch module inputs.
5. Routing display requests to the appropriate display supervisors.
6. Providing the flexibility for initializing the system in any mission phase.
7. Various timing functions.

Digital-to-Television

The Digital-to-Television control program provided a capability to monitor 300 independent displays on 28 displays channels. In addition, it provided displays from an auxiliary computer on four additional channels. A flight controller could view a display by directly requesting that display or by selecting the channel to which the display was assigned.

Group Display Control

The Group Display control program generated the data for displays, consisting of projection plotboards and X-Y paper plotboards in the Mission Operation Control Rooms and the control areas. It was designed to permit the system to function effeciently by assuming control of display processing and thus prevent congestion of computer facilities.

Manual Entry Device Control

The Manual Entry Devices (MED's) permitted users to control the Mission System program. A user could add, delete, or modify specified parameters and initiate certain processing. For example, the Docking Initiate MED entry initiated processing for calculating a set of maneuvers to effect a rendezvous of two vehicles. The Mission Supervisor routed the MED entry to the proper subsystem for decoding and execution.

Restart Procedure

The processing done by the Mission Operational Computer was backed up by the Dynamic Standby Computer. The primary processing function could be transferred quickly to the standby in case of malfunction or for preventive maintenance. This called for a procedure and control logic to load all the programs into a third computer that would assume the Dynamic Standby function. The Restart procedure included manual and control program actions that wrote the mission computer's storage contents on magnetic tape for loading and initializing a new standby computer.

OPERATIONAL READINESS AND CONFIDENCE TESTING

The Operational Readiness and Confidence Testing (ORACT) program system was designed to run independently or as part of the Mission System. It provided information on the operational readiness of equipment in the Houston Mission

Control Center and the Launch Trajectory Data System. This was done by comparing incoming data with expected data or by transmitting predetermined data for visual evaluation. In either case, an ORACT test determined the operational readiness of those portions of the Ground Operational Support System being tested and provided some degree of problem isolation.

SIMULATION PROGRAM SYSTEMS

The Ground Support Simulation Computer (GSSC) program system was part of the Simulation, Checkout and Training System (SCATS). The SCATS was designed to provide realistic simulations of the Gemini missions for training the Mission Operational Control Room teams, the remote site teams, and the astronauts. Models of operation were such that training exercises could be run for each group, combinations of groups, or a complete integrated simulation using all three groups.

During a simulation exercise, the group or groups participating performed their functions as if they were in a real mission situation. The SCATS created the simulated mission environment by generating and transmitting operational radar tracking data, real time telemetry data, command data, and various combinations of these and by reacting to inputs from the participants.

An important element of the SCATS was provided by the simulation controllers. They performed different functions than those of flight controllers in that they observed the progress of the simulated mission and intervened when necessary to bring about events for training purposes. An example would be to create an improper booster burn during a simulated launch to test abort procedures.

Simulation exercises for each mission were planned to include the nominal mission plan plus as many contingencies as time and resources would permit. In general, all data generated within the SCATS for transmission to the operational system could be biased, terminated or garbled at the simulation controller's discretion. With this capability, the SCATS provided training and testing of the operational support crews, equipment, and computer programs. Support for the SCATS at the RTCC was provided by the Simulation Operational Computer and the Ground Support Simulation Computer. Figures 6 and 7 compare typical mission and simulation configurations.

The GSSC was an IBM 7094 functionally assigned within the RTCC. The GSSC computed and transmitted the following on a real time basis:

a. TV displays to keep the simulation controllers aware of the current status of the simulated vehicles, network and related interfaces.

b. Simulated radar tracking data for transmission to the operational system.

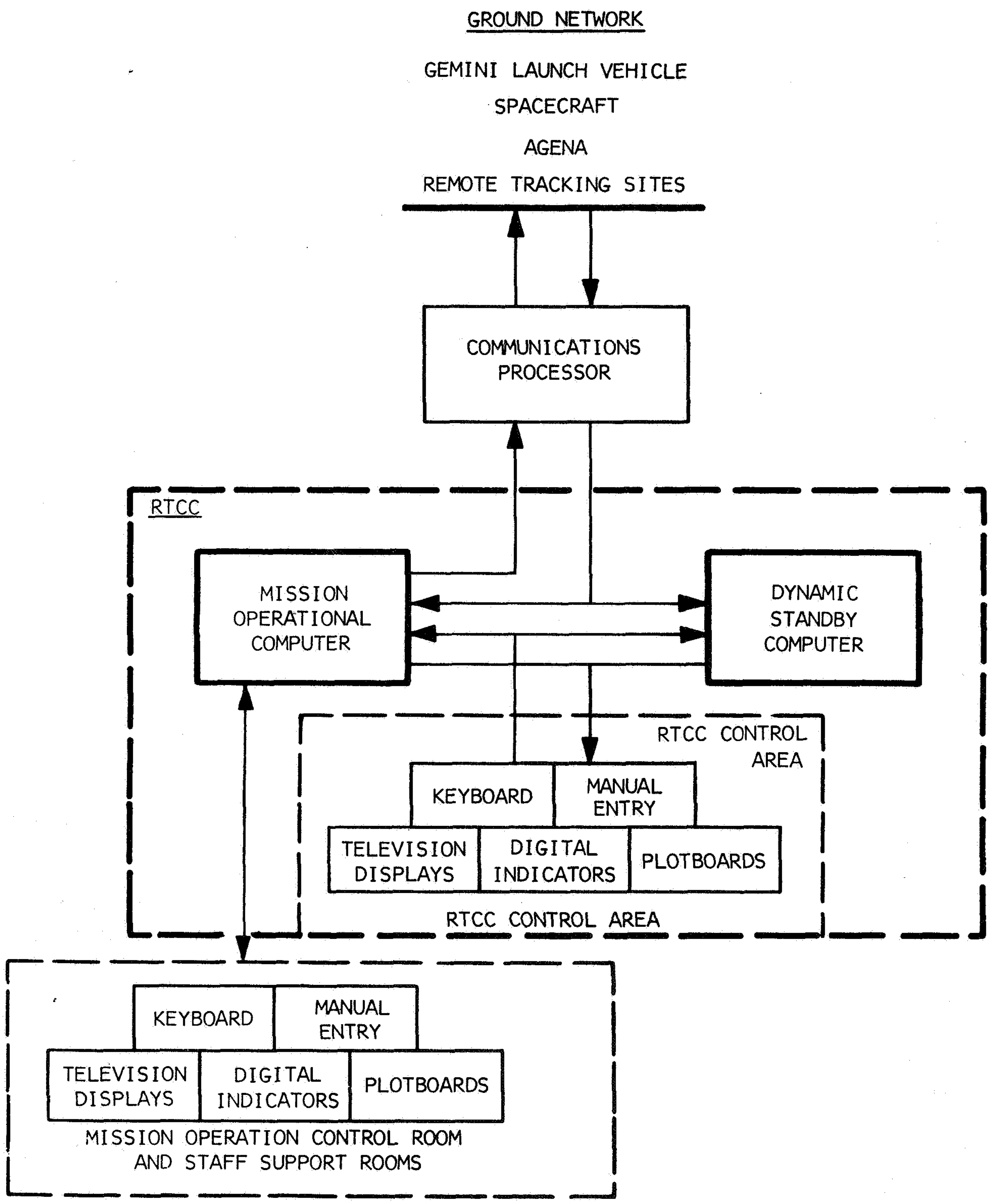


Figure 6. Mission Configuration

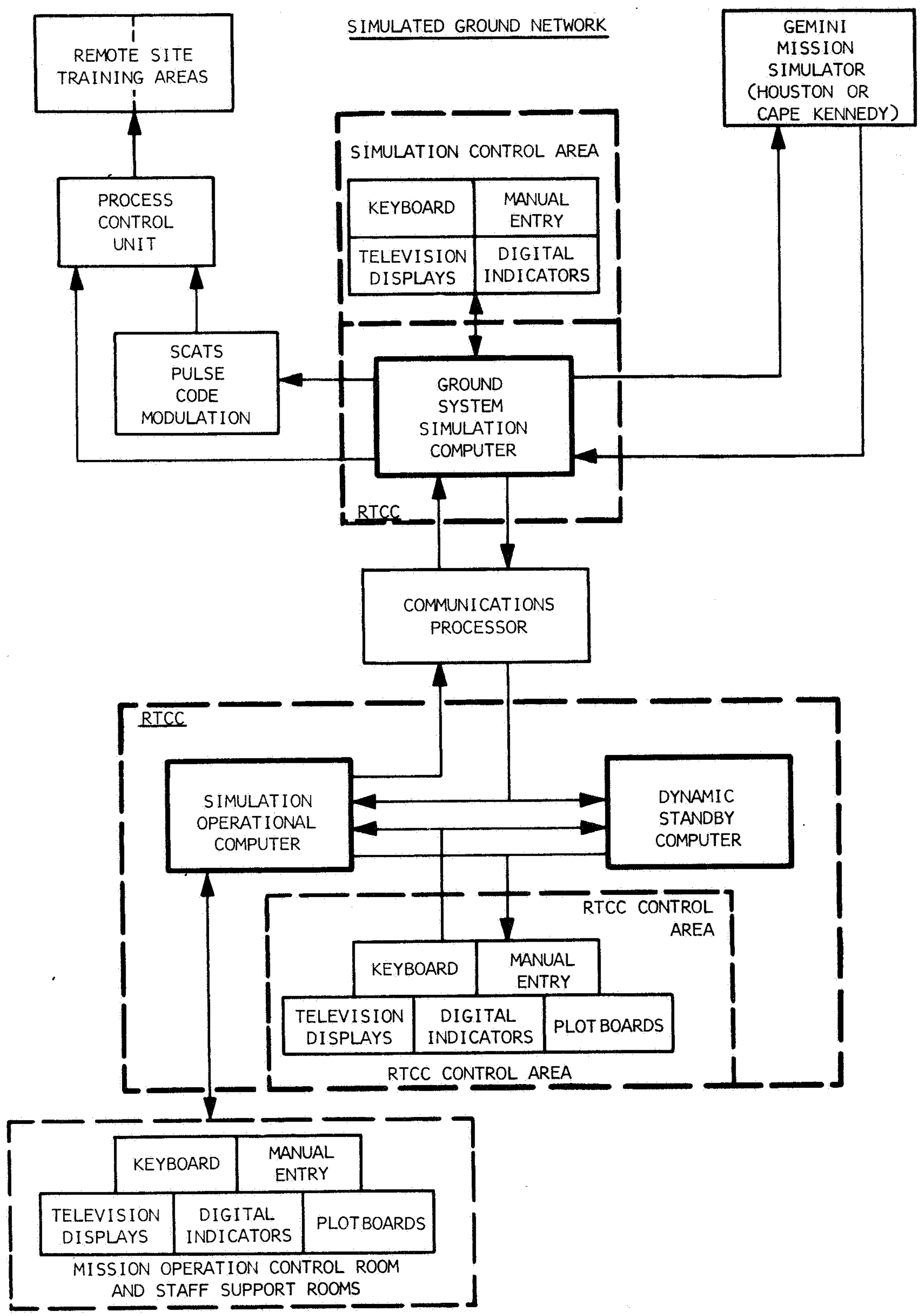


Figure 7. Simulation Configuration

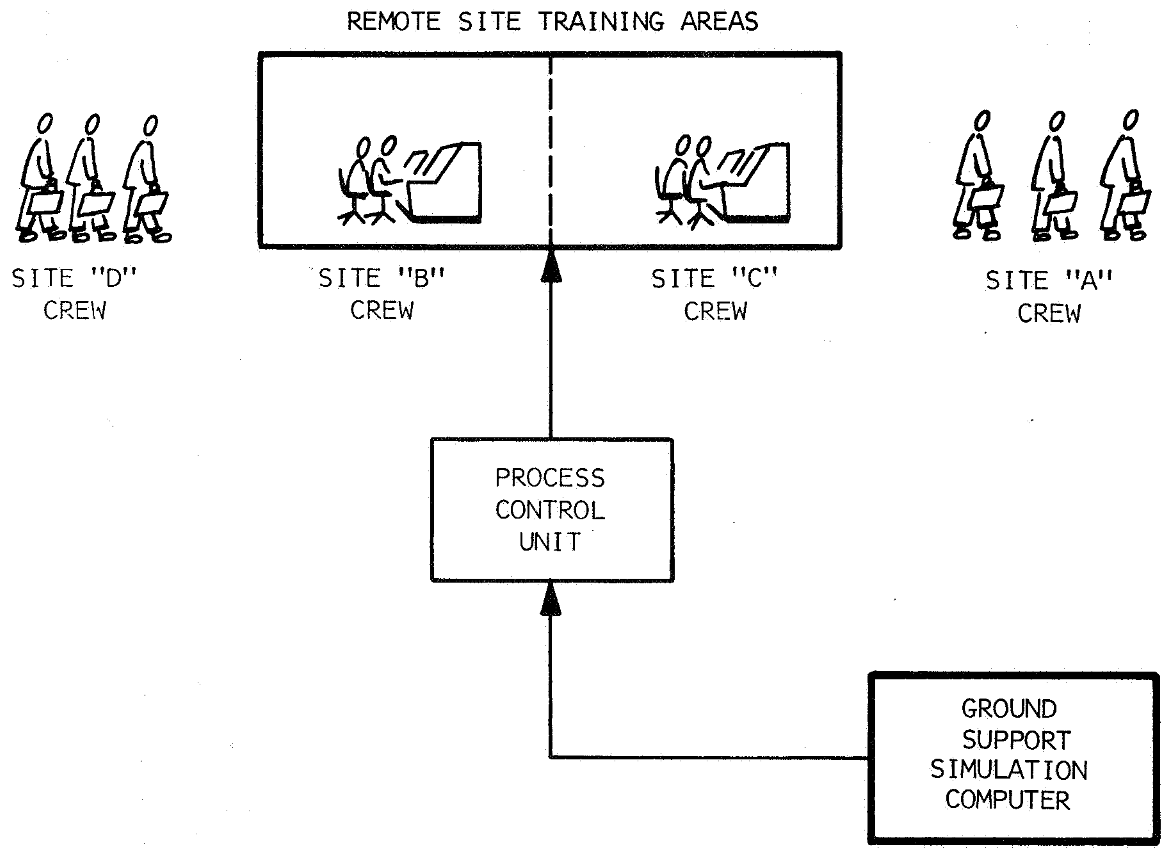


Figure 8. Remote Site Crew Training

c. Command control matrix for use by the Communications Processor in routing mission controller commands to simulated remote sites.

d. Simulated targeting parameters for the Gemini launch vehicles. These were originated by the Gemini Mission Control Facility at Cape Kennedy during the prelaunch and launch phases of an actual mission.

e. Control of remote site crew training. Two training areas at the Manned Spacecraft Center simulated remote sites. The GSSC assigned crews to these areas as data acquisition began and ended for each site during each orbit, and provided information to control routing of simulated telemetry data to the appropriate training area (see Figure 8).

To perform these functions, the GSSC computer program included:

a. An Agena vehicle model with telemetry, command response, "fault" capability (the ability to create non-nominal flight conditions), ground-track generation.

b. A Gemini vehicle model which provided generation of realistic telemetry in real time and which could be faulted, could respond to digital commands in real time, and could generate nominal or non-nominal trajectories.

c. Interfaces with Gemini Mission Simulators at Houston and Cape Kennedy for astronaut training.

d. Response to manual inputs from the controllers in the Simulation Control Area such as radar and telemetry faults, radar tracking controls, telemetry controls, vehicle model controls, remote site sequencing controls.

e. Starts and restarts in orbit for simulating a particular mission phase.

Just as an actual flight began in the prelaunch phase, the GSSC started its simulation with the transmission of 'pad' data. During this phase, GSSC played the role of the Guided Missile Control Facility by computing and transmitting the required guidance updates. During the Launch phase, the GSSC transmitted the data that would normally be received from various sites of the network based on the given launch azimuth. In this phase, the GSSC again played the role of the Guided Missile Control Facility. It did this by computing and transmitting the Radio Guidance System commands and Inertial Guidance System updates normal to the second stage burn of the Titan.

Whether the next phase was Orbit or Abort, the GSSC computed and transmitted the required data or, as in some type of abort simulations, stopped transmission of all data. During the reentry simulations, the GSSC again transmitted all the required data.

Two systems were delivered to NASA, the Gemini Mission Simulator Response package and the Gemini Launch Vehicle package.

The Gemini Mission Simulator Response package could interface with either the Gemini Mission Simulator at Cape Kennedy via the Communications Processor or use prerecorded, fixed profile input data tapes. It contained all the logic required to support all functions except those that required the Gemini math portion of the Gemini Launch Vehicle package. In addition, this system contained the Atlas/Agena mathematical model which for any rendezvous mission was used to simulate the first vehicle in orbit. The development of the Atlas/Agena math model as a training aid for a real time environment represented a significant milestone. The model represented a complete digital simulation of the vehicle as opposed to standard methods. The Atlas/Agena vehicle model contained closed loop guidance for the Launch phase and simulated the following Agena systems:

a. Guidance and Control System

b. Primary Propulsion System

c. Secondary Propulsion System

d. Electrical Power System

e. Telemetry and Communication System

f. Digital Command System

The model generated 392 telemetry parameters.

The Gemini Launch Vehicle package could simulate the Prelaunch 2 phase, a guided launch with variable orbit insertion conditions, the nominal orbital attitude and maneuver system insertion burn (a burn to achieve desired orbit conditions), and an Abort phase. To accomplish these objectives, mathematical models were developed which simulated the Titan II booster and the Gemini spacecraft with their associated systems. These models were designed to simulate nominal and non-nominal conditions. Capabilities were provided to respond to system and telemetry sensor fault insertion, Digital Command System commands, and simulation controller inputs. The GLV/Gemini models simulated concurrently the primary radio guidance and the secondary inertial guidance systems. Redundant hydraulic and electrical power systems were also simulated to provide realism and effective training for the system switchover exercises. The GLV/Gemini models generated 488 telemetry parameters.

The GLV package simulated the following:

a. Propulsion and Pressurization System
b. Electrical Power System
c. Sequential System
d. Malfunction Detection System
e. Guidance and Control System

The Gemini package simulated the following:

a. Sequential System
b. Control System
c. Inertial Guidance System
d. Orbit Attitude and Maneuver System
e. Communication System

The GSSC program system was highly complex since it had to satisfy many conditions for each of the various types of simulations. Flight controller training exercises were run with the Gemini Mission Simulator (GMS) at Cape Kennedy, with the simulator at Houston, or with programmed mathematical models in place of the GMS. It could operate with or without remote site crews. In addition, the program system could simulate all or any part of the particular Gemini mission for which it had been designed. As many "faults" as possible were programmed into the system to test capabilities and train crews to react to any contingency during a mission. For instance, the GTA-12 Atlas/Agena vehicle model was designed to respond to 149 different fault requests, and the Gemini Launch Vehicle/Gemini model to 90 fault requests.

RTCC OPERATING SYSTEMS

The present state of computer technology and the processing requirements of a real time system made operating systems a vital part of the RTCC. Operating systems are groups of programs that control a system with continuous operation over a span of many jobs and by providing a library of utility programs.

Real time data processing would obviously be impossible to perform if manual operator intervention were required to recognize and initiate processing for each particle of data. Operating systems provided the RTCC with the ability to respond to real time requirements with increased efficiency of operations. The concept applied in the RTCC recognized the difference between application programs, which solve problems, and control programs, which perform general processing control. Thus, the control program directs and coordinates the execution of the application program.

Virtually all processing at the RTCC operated under the control of either the Executive Control Program or the Compiler Operating System. The following concepts were followed in the design of these operating systems:

a. Separation of application program logic and control logic. The Executive was independent of any specific application logic, providing the ability to change one without affecting the other.

b. Adaptability of the total equipment and program system, permitting the system to function in a great variety of circumstances.

c. Project-wide use of the same control system.

d. Use of as many standard IBM commercially available programs as possible.

e. Modularity of programs with communication between application programs through standard control program service routines.

f. Independence of application programs from the equipment configuration, so that configuration changes would not affect the application programs.

The Real Time Executive Control Program was basically a 13,000-word program that resided in the computer during RTCC operations. Executive provided for priority-based multiprogramming, dynamic storage management and allocation, data control and routing, dynamic statistical monitoring, protection from memory overload, real time error recovery and restart, and communication between programs. These services were available in an environment sufficiently responsive to monitor manned space missions in real time. In addition, the system offered comprehensive program library management facilities.

The Executive Control Program design was coordinated with the planning of application program design. Early in the project, RTCC programming groups adopted the use of two types of application programs for RTCC systems. These were processors and supervisors. Supervisors were programs that contained the logic for a specific application function. Processors actually performed the detailed functions under the direction of supervisors. Supervisors could work together to perform a function. For instance, low-speed radar messages were received and stored under direction of the Low Speed Radar Supervisor, timed under the Orbit Time Supervisor, and processed under the Differential Correction Supervisor to calculate a position and velocity for the space vehicle.

The supervisors made continuous use of processor programs. Many standard mathematical functions (e.g., square root calculations) and other common routines were written as processors. Since these could be used by any supervisor, greater efficiency and standardization were obtained. Supervisors could communicate with processors and other supervisors, while a processor could communicate only with the supervisor that "called" it.

Supervisors and processors performed their functions under control of the Executive Control Program. All incoming real time data was received by Executive. Based on predetermined routing directives from the supervisors, Executive stored or routed the data to the required place in computer storage and signaled the supervisor when it was to begin processing. When the supervisor was ready to assign work to a processor, it "called" the processor by requesting Executive to transfer data and computer control to the processor. At the completion of the processor's task, control and results of the computing were normally returned to the supervisor by Executive.

Executive could move programs in and out of main storage depending on predetermined rules and priorities. Executive could interrupt the processing of a program to permit a higher priority activity to occur. It would later return to the interrupted program. Also it controlled all system output according to the type of requirements, such as the output device and data format.

Several unique operating features were provided by Executive to take advantage of the design features just described. Those aimed at program development simplified the integration, testing and debugging of RTCC systems. Applications supervisors and processors were built into units. The units were tested and built into subsystems which were perfected before or during the building and testing of a complete system. This process reduced the length of time required to produce an operational system and permitted the programmers to specialize in specific applications. It also provided great flexibility for adding functions or modifying existing ones, since the chances of one change upsetting the entire system were greatly reduced.

For real time development and operation, Executive provided a means of system initialization. Initialization routines created the required environment in a system to enable processing to begin. At the direction of controllers or programmers, the computer operator entered initialization options, such as the mode of operation (real time simulation, operational), details of processing requirements, and the desired equipment configuration.

Multiprogramming, or processing several tasks at once, was another capability of Executive. This improved responsiveness and adaptiveness to a variety of input data and output requirements.

During the Gemini missions, Executive was keyed to keeping the real time system running no matter what conditions were encountered. In doing this, Executive provided an elaborate error recovery system to intercept errors in the program, hardware, and data. When one of these errors were encountered, Executive provided an appropriate recovery method to enable the real time processing to continue. If the error encountered was of such a nature that recovery was either impractical or unfeasible, Executive recommended switching to a standby system (see Restart Procedure, page 25). The 65, 000 words of main core storage and the 524, 000 words of auxiliary storage could then be transferred to a new standby computer in less than five minutes by the Executive restart logic. If an input or output device failed in any manner, Executive provided logic to ensure that failure of one device would not interfere with processing in the rest of the system.

Several simulation and debugging techniques were made available by Executive. A Manual Entry Device Simulator accepted manual entry requests in punched card form. Testing could then take place when Manual Entry Device equipment was not available. The Simulated Input Control routine and the Statistics Gathering System, explained elsewhere in this report, were part of Executive. The debugging package consisted of many aids to debugging, such as printouts of all or part of computer storage during program execution, viewing of any part of storage on a television display device, and synopses of storage printouts.

Compiler Operating System

In addition to program system design and applications programming, a means was required to build and operate each system. To do this the standard IBM 7094 operating system (IBSYS) was modified to meet the special requirements of the RTCC. This system was called the Compiler Operating System.

A major design effort was involved in creating the Compiler Operating System. The system provided the basic requirements necessary in any data processing effort to make a programming language available and to compile and assemble programs written in this language. The most significant contribution made in this area was a feature known as the Gemini-Apollo Editor. The Gemini-Apollo Editor provided the ability to build, maintain, and modify real time systems on master tapes both in source (programmer) language and binary (machine) language. This permitted integration and testing of subsystems as they were developed in the construction of a master system in a "building block" fashion. This capability was further aided by a program developed for the RTCC called the Job Shop Executive Simulator. This program was made available during early development to permit simulated real time testing with only a partially built system to perform early testing of completed programs.

Special debugging aids as well as several technical features designed to create more economocal system development also were provided by the Compiler Operating System.

EQUIPMENT SYSTEMS

The Real Time Computer Complex consisted of computers and associated special equipment to provide data processing and to interface with real time input, output, and display systems. This equipment was used to support mission and simulation functions as well as to develop the programming systems.

The original complex was three IBM 7094 "subsystems," each having 32,000 words of main storage and 98,000 words of auxiliary storage. It grew to five IBM 7094 Model II subsystems with 65,000 words of main storage and 524,000 words of auxiliary storage.

Whenever possible, IBM recommended standard commercial equipment. It was necessary, however, to provide some special units to fulfill unique requirements. In these cases, the equipment was either rented (as with logical extensions to the capability of existing equipment) or purchased (as with special equipment designed and built solely for RTCC).

FUNCTIONS

The five IBM 7094-II computers functioned independently, or "standalone," and could be assigned any of the following functions in the RTCC.

a. Mission Operational Computer (MOC). This computer was loaded with the Mission Operational Program System and acted as the real time processing unit during mission status.

b. Dynamic Standby Computer (DSC). This computer was also loaded with the Mission Operational Program System and operated in parallel with the Mission Operational Computer except that it transmitted no functional output.

c. Simulation Operational Computer (SOC). This was a computer loaded with the Mission Operational Program System to provide processing during a simulation exercise. It was not backed up with a standby computer.

d. Dynamic Network Data Generator (DNDG). The Dynamic program was loaded into this computer for use in testing the Mission Operational Program System. It operated with the Mission Operational Computer.

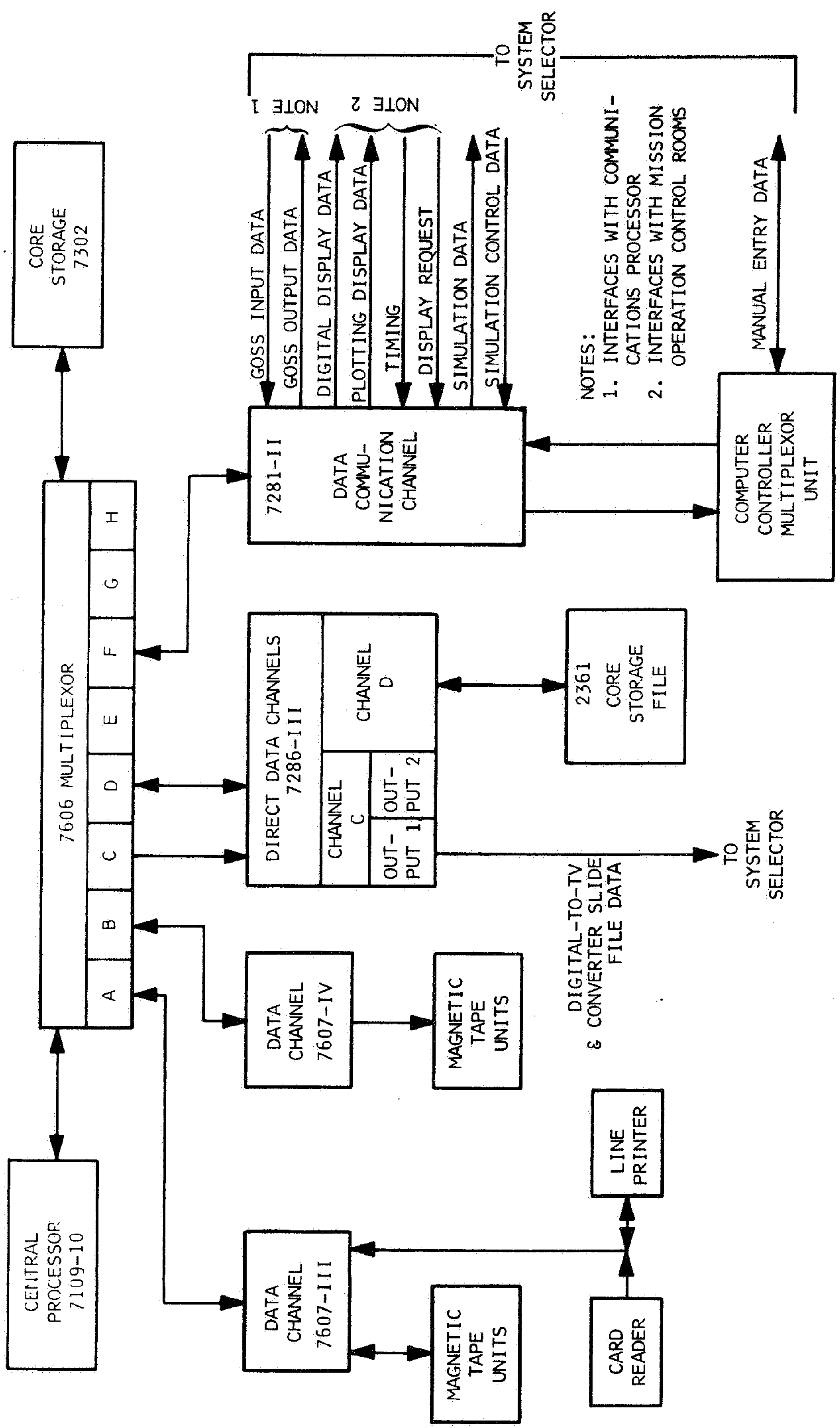


Figure 9. RTCC Computer Subsystem

e. Job Shop Computer. This was used for computer program generation or program study and evaluation. It was normally used as a batched job or non-real time processor.

f. Ground Support Simulation Computer (GSSC). This computer operated with the GSSC program system in support of the Simulation, Checkout and Training System to provide simulated real time mission data for training flight controllers and astronauts and for checkout of systems.

In a mission configuration, one computer subsystem was assigned as MOC and a second as DSC. It was possible to configure simultaneously for two missions, a mission and a simulated mission, or two simulated missions.

COMPUTER EQUIPMENT

Selection of the IBM 7094-II and the design of the RTCC computer subsystems emerged from analysis of the requirements imposed by the real time programming systems. The highly sophisticated, modular programs and the real time functions they supported required a computing system with the capabilities of the 7094-II.

The five identical 7094-II computers each consisted of the main storage and computing units, data and communications channels to interface with the outside world, magnetic tape storage units, high-speed printers and readers, and consoles for operator communications (see Figure 9). The computers provided the following capabilities.

a. High-speed internal processing.

b. Access to large storage areas at high data rates.

c. Acceptance of large amounts of real time data and control information.

d. Outputs of computer-generated displays in near-real time.

The computer handled data internally in a fixed word length of 36 binary bits. Information was transferred in full-word parallel form. The computer cycle time, or the time it took to perform one logical arithmetic or transfer operation, was 1.4 microseconds. This cycle was determined by the magnetic core storage in the 7094 which was a fast, direct-access device. A 36-bit data word could be read into (or out of) any one of its 65,536 storage locations in 1.4 microseconds, or one computer cycle time.

Each of five input/output channels operated as an independent unit to transfer data to and from core storage; however, each had to be initiated by and remained under the direction of the control program. Once set in operation by the program, the channels continued uninterrupted until the operation was complete. Data was transferred by the channels to and from storage in parallel form, or one full word at a time. The specific functions of the channels are described below.

a. IBM 7607 Data Channels - Each of the two IBM 7607 Data Channels connected a bank of six magnetic tape units to the 7094. One channel had the additional function of providing a connection for the printer and card reader.

b. IBM 7286 Direct Data Channel - The 7286 contained two independent channels. One channel provided the necessary interface for operating the Large Capacity Storage with the 7094. The other transmitted digital-to-television data and display coding.

c. IBM 7281-II Data Communication Channel - The 7281-II was a multiplexing device. This unit connected one 7094 channel to 13 separate subchannels.

The IBM 7281-II included five input and six output data subchannels, one clock subchannel, and one interval timer subchannel. The input and output subchannels interfaced with external devices, such as the Communications Processor and devices in the Mission Operation Control Room. The subchannels transformed data from internal 7094 full-word parallel transfer to the bit-by-bit serial data transfer of the external devices and vice versa. The interval timer subchannel provided a means of signaling, or interrupting, the computer when a predetermined interval of time had elapsed. The clock subchannel interrupted the program at unit intervals of time at a program-selected rate.

An IBM 2361 Large Capacity Storage was part of each 7094 subsystem and provided 524,288 words of auxiliary random-access core storage. The computer program would store or fetch large amounts of data at a rate of 250,000 words per second, making it convenient for keeping programs (and their associated data) available to the mission control program for fast recall into the 7094's computational storage.

A Computer Controller Multiplexor Unit was associated with each computer to transfer data between the 7094 and manual entry devices and switch modules in the RTCC Control Area.

COMPLEX EQUIPMENT

Specially build units were included in the computer complex to perform special functions and provide real time interfaces. These included the Time Standard Unit, the Standby Digital Driver Unit, RTCC Control Area devices, and the System Selector Unit. The Time Standard Unit generated accurate timing signals for various timing needs. The Standby Digital Driver Unit was used to monitor the computations of the Dynamic Standby Computer to assure its readiness.

Control of the RTCC system was lodged in the RTCC Control Area adjacent to the computers. It consisted of two groups of display consoles manned by IBM and NASA controllers. The purpose of the Control Area was to monitor the performance of the computers, to examine the quality of input and output data, and to provide direct support to the flight control teams in the Mission Operation Control Rooms. Various specially designed equipment was used in connection with the Control Area to provide displays and interfaces with the computers.

The assignment of the RTCC computers to any of the various functions, such as Mission Operational Computer, Dynamic Standby Computer, or Dynamic Network Data Generator, was accomplished by the System Selector Unit (SSU). The System Selector Unit also provided the capability to rapidly execute the changeover between the Dynamic Standby and Mission Operational computers during a mission. This could be performed within two-tenths of one second.

Control of the RTCC configuration by the System Selector Unit was accomplished by patchable plugboards and relays. Each of the five IBM 7094-II computers was connected to the unit. The unit, in turn, provided paths between the computers and the various external display, control, and communications equipment. Hence all real time input or output of any computer went through the unit. Certain plugboards were prewired for the more typical configurations.

An example of a typical configuration could have been one used to run simultaneously a simulation exercise, a dynamic program test, and job shop operations (see Figure 10). Two computers would be required to run the simulation exercise. One would be designated the Ground Support Simulation Computer, and the other the Simulation Operational Computer. The GSSC would be connected via the System Selector Unit to the Communications Processor and also to the Mission Operational Control Room where simulation controllers would be stationed. Thus, simulated telemetry and trajectory data would originate in the GSSC, go through the System Selector Unit to the Communications Processor for conversion to RTCC

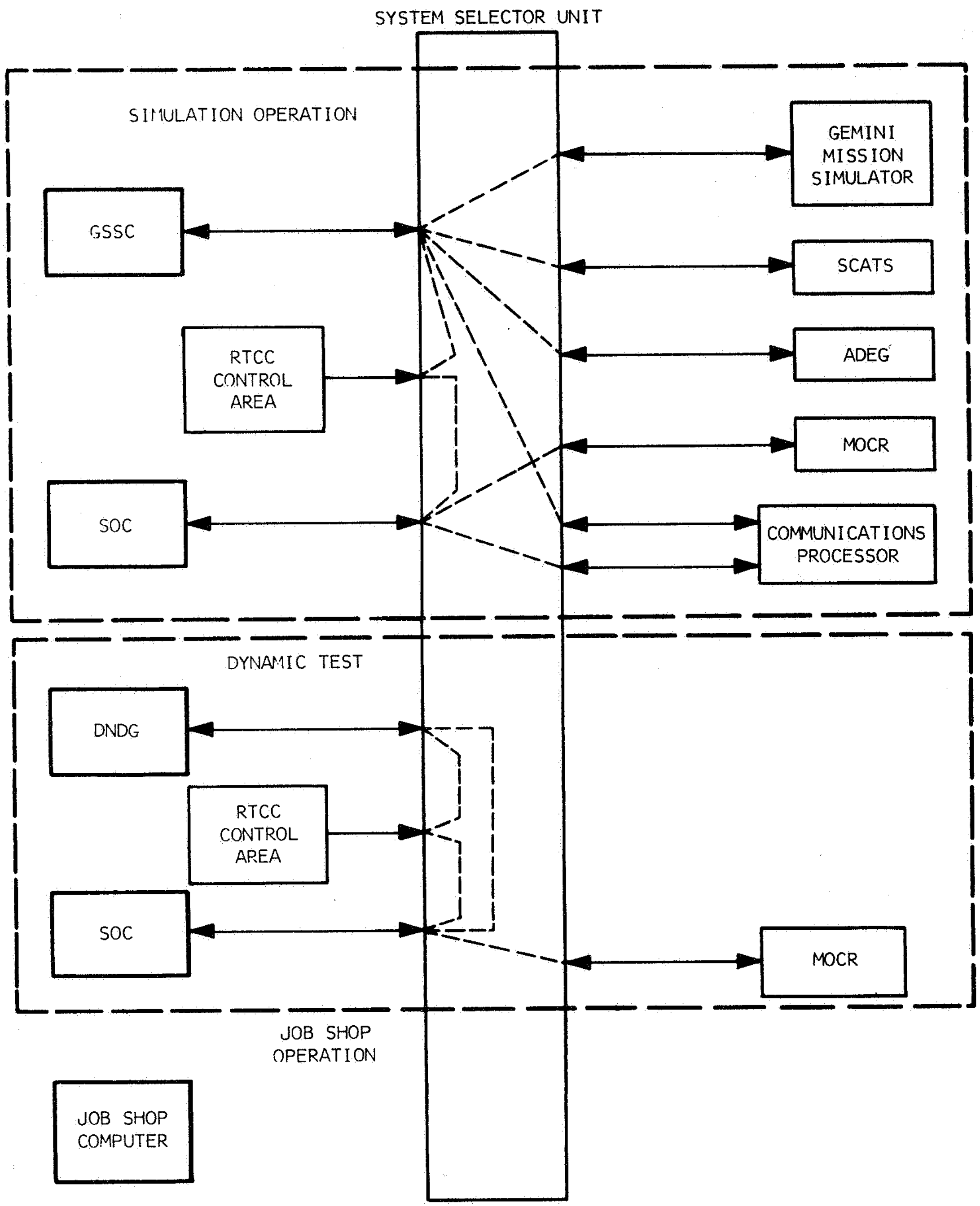


Figure 10. System Selector Unit Operation

input format, go through the SSU to the Simulation Operational Computer, which would process it and pass the results back through the SSU to controller consoles. Controller responses would follow the same route in reverse, again passing through the SSU three times.

The program testing exercise would also involve two computers. A Dynamic Network Data Generator and a second Simulation Operational Computer would be coupled through the System Selector Unit. The Mission Operational Program System or a part of it could be tested in this manner.

The fifth computer in this example could be compiling or testing programs for future missions. This job shop computer did not operate in real time and would not be connected to the System Selector Unit. Its input would be from magnetic tape or punched cards, and outputs would be on tape and printed listings. Increased efficiency in job shop operations was obtained by the use of peripheral processing systems such as the IBM 1401 computer. The 1401 was used mainly to convert job shop input data from punched cards to magnetic tape and output data from magnetic tape to printed output. Thus, the IBM 7094 could process this data more efficiently with high-speed magnetic tape and was freed from the slower punched card and printer operations.

LAUNCH TRAJECTORY DATA SYSTEM

A specialized equipment system, the Launch Trajectory Data System (LTDS), was designed for use during prelaunch and launch phases of missions and simulated missions. The LTDS contained equipment and data flow paths for processing and transmitting launch trajectory data acquired by the Air Force Eastern Test Range and Bermuda site radars to the RTCC at Houston. It also provided for transmitting trajectory data, merged with selected telemetry events data, from the Cape Kennedy Impact Predictor and Guided Missile Control Facility sites to the RTCC. Telemetry events data were supplied to the LTDS from telemetry ground stations located at Cape Kennedy.

In addition, the LTDS provided for transmitting computer prelaunch data from the RTCC to the Guided Mission Control Facility computer. Design requirements stipulated that the system support missions from the start of transmission of computed prelaunch data until the completion of trajectory data transmission from the Bermuda or Antigua radar sites.

The LTDS also provided the data path and interconnection facilities between remote mission simulators (at Cape Kennedy and the Manned Spacecraft Center) and the simulation subsystem of the RTCC during simulation exercises.

The data transmission equipment used a cyclic error-detection code technique which provided an extremely reliable means for message synchronization as well as detection of data circuit transmission errors. The error code generation function was performed by a Data Control Unit Transmitter at the source terminal and was decoded by a Data Control Unit Receiver at the receiving terminal. A total of seven Data Control Units were installed at the Manned Spacecraft Center and the various other sites to permit two-way communication and flexibility during LTDS operations.

Since the LTDS provided operational support during one of the most critical phases of a mission, it had to perform with a high degree of reliability. Duplexed facilities, therefore, were provided for each of the trajectory data links from Cape Kennedy and Bermuda to the RTCC. Extensive switching facilities were included to permit the interchange of equipment or transmission circuits to continue operation in the case of failure of portions of the two data paths.

DEVELOPMENT MANAGEMENT, SUPPORT AND OPERATIONS

The coordination of planning, program development, equipment development, and operations is vital to successfully fulfilling contract obligations, especially in the dynamic environment of a research and development project. Throughout the Gemini program, IBM and NASA continuously discussed problems, experimented, innovated, and adapted to change concerning management techniques and technical development. Some of the methods used in these efforts are described in this section.

The organization was structured around the development and operation of mission support. A functional concept was retained, although the organization changed several times due to growth or to answer special needs and changing requirements. The functional division of work afforded concentration on several areas of development of both program and equipment systems. At the same time, it provided the operational capability required to fulfill the mission support function.

In addition to the direct support of IBM personnel at the Manned Spacecraft Center, the resources of the entire IBM Corporation stood behind the quality of systems installed at the RTCC. The IBM Data Processing Division and Field Engineering Division services in Houston as well as various IBM efforts in program development, manufacturing, and product design provided invaluable support to the RTCC project.

Close communications between IBM and NASA came about in several ways. Perhaps the most significant example was the introduction of the development plan. Development plans were the basic management tools used in planning, reviewing, and reporting the technical aspects of development and testing activities. Physically, a development plan was a document which contained the current plans and status of an area of effort. The Equipment Systems Development Plan encompassed all aspects of special and standard commercial equipment planned and installed. Development plans existed for each deliverable program system throughout the period of its development. The format of each type of development plan was tailored to the particular nature of its subject. The successful use of the development plan stemmed from the thoroughness of the information and the attention it received. All requirements under development as well as all changes under consideration or development were included. Each development plan was reviewed weekly by IBM management to study progress, discuss problems in detail, and publish a revised plan. A weekly review meeting was then held with IBM management and NASA technical monitors to discuss the plan. Thus, the development plan became a written documentation of IBM's plans and status for each deliverable item.

In addition to development plans and informal communications, several other types of documentation and reporting techniques were provided by IBM as required in the statement of work. These included programming and equipment documents, periodic and special reports, and procedural plans.

Advanced techniques were needed to analyze system designs and performance. The IBM General Purpose System Simulator (GPSS) and the Statistics Gathering System (SGS) proved to be highly effective for these purposes. The GPSS is a simulation program which was used to create models describing the logic of all or parts of equipment and program systems. The models were built using estimates of the execution, timing, and logical characteristics of programs in a real time system. This permitted continual study of systems from early design concepts through their development and testing, well in advance of their use. An orderly and reliable process for evaluating combinations of various design alternatives resulted. These studies were used to assure adequate mission support at the least cost.

The Statistics Gathering System (SGS) was specially designed by IBM for RTCC use. It was a subset of the Executive Control Program for use during testing and simulations to provide timing information and statistics on the frequency of program execution. Therefore, what the GPSS could learn about planned systems, the SGS could learn about existing systems. This provided a way to analyze the performance of current real time systems. SGS was also used to evaluate the accuracy of GPSS model predictions by "rerunning" the RTCC input data recorded during an actual mission. GPSS predictions were generally found to be within 5 percent of actual performance.

Plans for quality assurance, testing, acceptance, reliability, and maintainability of equipment were established early and were followed throughout the course of the project. The equipment testing technique evaluated the initial completeness of design requirements and the continuing ability to meet these requirements. The progression of testing from the basic unit level through subsystem and system levels provided the basis for special equipment acceptance by NASA. The acceptance was the documented recognition of the ability of the equipment to perform its specified functions. The test specifications served as the standard for determining this ability. Standard data processing systems were initially measured against the General Services Administration's standards of performance and acceptance. After installation, more detailed testing of the RTCC system continued throughout the Gemini project.

One of the most difficult and time-consuming aspects of program development lies in performing adequate, comprehensive tests of the program system's capability to perform intended functions. The basic purpose of computer program testing is to ensure its ability to perform its intended functions at the time those functions are needed. A "functional" testing technique was developed and incorporated into standardized formats for test specifications and reports for all levels of test activity. (Functional testing involves the execution of somewhat independent experiments designed to demonstrate or verify a specific functional capability of the system.) Three levels of testing were established to ensure that an adequate examination of the program was made at the micro, intermediate, and macro level of detail. Thorough testing of the basic building blocks prior to their integration into large component parts of the system greatly reduced the number of problems encountered at the next building and testing step.

Due to the critical nature of the Mission Operational Program System, a special testing technique was devised. To provide a complete checkout of the Mission System, the Dynamic Network Data Generator (DNDG) program system was designed. It presented a real time simulation of all tracking and telemetry data received by the mission computer of the Agena and Gemini vehicles and to a limited extent of the Guided Missile Control Facility. When the system was in operation, the basic computer configuration was two IBM 7094's with or without the Communications Processor. The DNDG program (referred to as Dynamic) resided in one IBM 7094 and the Mission program in the other. All trajectory and telemetry data were normally routed from the Dynamic to the Mission program directly, but they could be routed through the Communications Processor. The Mission program received and transmitted the data just as it would during an actual mission, and all phases were simulated. Flight controllers manned their consoles as in a mission. The Dynamic program responded to Mission program outputs in an actual manner. These simulated missions could start or restart in any phase desired, and at any point between phases. A four-computer configuration could be used for checkout of restart and changeover.

Control information was entered into the Dynamic program by an operator at a console in the RTCC Control Area via a Manual Entry Device. Examples of this information were length of prelaunch, insertion conditions, maneuvers, and data faults. Radar and telemetry data for the two vehicles were generated by various subprograms within the system.

The Dynamic program was designed to present a realistic mission in real time during all phases with complete flexibility and with capabilities to introduce system "bugs" or faults. The ability to start or restart a mission in any phase facilitated checkout of the Mission program by enabling testing of the various packages independently or jointly. The capability of faulting and controlling in real time all data generation and transmission permitted checkout of selected mission functions. The capability of responding realistically to commands presented by the mission computer further aided in assuring complete checkout of these extremely vital functions.

The Gemini Dynamic-Script system was designed to simulate the Gemini and Agena vehicles in all the proposed phases of flight. High- and low-speed radar and telemetry data were generated for all the tracking sites. Two modes of testing were available: real time data generation of Dynamic, and non-real time or "Script" testing. In the Script mode, the data contained on a previously prepared tape was fed to the Mission program via a Simulated Input Control (SIC) program. The Script mode had two distinct advantages over the Dynamic mode of testing. First, the mission computer ran faster than in the Dynamic mode; and second, only one computer was required. Both of these benefits provided an obvious cost saving. Various types of data faulting were also available in this mode.

The system test activity was followed by operational support functions. Test teams were composed of members of IBM and the NASA Flight Software Branch. These teams also served as operational support teams for simulation exercises and missions. Team members were called on to make quick and accurate evaluations of complex situations.

A demanding though unheralded task lay in performing the maintenance and operation of the many elements of the computer systems and special equipment. Preventive maintenance plus the installation of new equipment and engineering changes were provided by a specialized team of experts to assure performance during mission and development activities. Twenty-four-hour availability of equipment operators and of engineers to service failures was required, even during periods when no mission support activity was going on, to meet delivery schedules and maintain standards of computer availability.

CONCLUSION

Project Gemini was a creative challenge to IBM. In less than three years the Real Time Computer Complex was developed and given its first test to prove it could support manned space flights. Since GT-4, the RTCC has undergone continual growth in capability, flexibility and sophistication. This growth has been reflected in the ability and confidence of the people who have served the project.

Although the descriptions of systems in this report have been referred to in the past tense, many are still in operation in support of the Apollo program. Those systems which are no longer in use are those which have been replaced by more recently developed and more powerful systems necessary to handle the increased demands of Apollo support. At the time of writing this report, two Apollo missions have been supported by IBM at the RTCC, and programming systems for future missions are under development. Various equipment changes are being made, most significantly, the conversion of computer support from the IBM 7094-II's to "third generation" technology of the IBM System/360 Model 75. Programming efforts have expanded to take advantage of the increased power of the System/360 and to create the more complex systems required by Apollo missions.

Thus, IBM's technology grows as its involvement in the NASA space program deepens, and its enthusiasm rises to meet the growing challenges of the future. The people of IBM appreciate this opportunity to assist the country's efforts to land an astronaut on the moon in this decade and continue the quest for benefits to man through the exploration of outer space.